

Università degli Studi di Trento Corso di Programmazione I – E3: 145430/145935 2° Appello – 06/02/2018 – a.a. 2017/2018		PC N°						
COGNOME ..... NOME ..... MATRICOLA <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> </tr> </table>								

Completare le voci **Cognome**, **Nome** e **Matricola**, se tali voci saranno mancanti o non-leggibili elaborato **NON** verrà corretto.

**NON** è consentito utilizzo di alcun strumento cartaceo, elettronico, informatico etc. durante la prova.  
 Fa parte della valutazione la leggibilità del codice C/C++.

**Scrivere le risposte su questo foglio e riconsegnare. I risultati verranno pubblicati su DOL**  
**Tempo a disposizione: 120 minuti.**

<b>PARTE TEORICA: 12 PUNTI – PUNTEGGIO MINIMO 6 PUNTI</b>
---

**[Domanda 1 - punti 4]** Dato un file di testo “persone.txt”, si supponga che sia costituito da righe ciascuna contenente una stringa (nome della persona) ed un intero (età). Ad esempio:

mario 31
lucia 28
gianni 22
....

Scrivere la porzione di codice in grado di i) aprire il file, ii) leggere il contenuto del file e iii) stampare a video i nomi delle persone che hanno l'età maggiore e minore.

**Nota.** Nel caso di valori multipli di anni massimi o minimi, considerare il primo.

**[Domanda 2 - punti 4]** dato l'algoritmo di MergeSort

```
void MergeSort (int A[], int p, int r) {
    int q;
    if (p<r){
        q=(p+r)/2;
        MergeSort (A, p, q);
        MergeSort (A, q+1, r);
    }
}
```

Indicare in ordine le chiamate effettuate nel caso di chiamata **MergeSort (A, 0, 3)**

Indicando esplicitamente i valori dei parametri formali con cui viene invocata la funzione.

**[Domanda 3 - punti 4] Descrivere un algoritmo che moltiplichi due interi,  $n$ ,  $m$ , senza usare la moltiplicazione o divisione e che permetta di riempire la tabella di sotto.**

**Qual è il migliore algoritmo che potete progettare, ovvero che abbia  $T(n)$  (funzione delle istruzioni da eseguire) migliore.**

$n, m$	1	2	3	4	5
1	1	2	3	4	5
2	2	4	6	8	10
3	3	6	9	12	15
4	4	8	12	16	20
5	5	10	15	20	25

**PARTE PRATICA: 21 PUNTI – PUNTEGGIO MINIMO 11 PUNTI**

**[A - punti 3]** Inserire i seguenti tipi in un file **dati.h** e implementare i metodi di stampa:

```
typedef enum Ttipo{ AUTO, MOTO, AUTOBUS } Ttipo;  
  
typedef struct Tveicolo{  
    char marca[20];  
    int cil;  
    float kw;  
    Ttipo tipo;  
    // metodo stampa  
} Tveicolo;  
  
typedef struct TcodaFIFO {  
    int n, dim, head, tail;  
    Tveicolo *s;  
    TipoCoda (int x) {  
        dim = x; n = 0; head = 0; tail = 0;  
        s = new Tveicolo [x];  
    }  
    // metodo stampa  
} TcodaFIFO;
```

**LEGENDA struct TcodaFIFO**

**n** = numero elementi nella coda  
**dim** = dimensione max coda  
**head** = posizione elemento in testa  
**tail** = posizione elemento in coda  
**s** = vettore elementi

**[B - punti 1]** Modificare il file **main.cpp** includendo il file **dati.h** ed inserendo il seguente codice nel **main**:

```
TcodaFIFO* p = new TcodaFIFO(10);  
Tveicolo v;  
for(int i=0; i<5; i++) { creaVeicolo(&v); addVeicolo(p,v); }  
stampaVeicolo(p);  
delAuto(p, 2000);
```

**[C - punti 4]**

Implementare una funzione **creaVeicolo** che riceve in input per riferimento una variabile di tipo **Tveicolo v** e la inizializza con:

- campo **tipo** valore casuale tra AUTO, MOTO, AUTOBUS;
- campo **cil** valore letto da tastiera, tra 900 e 4000, con controllo input.
- campo **kw** pari ad un diciassettesimo (/17) del valore **cil**;
- campo **marca** letta da tastiera.

**[D - punti 5]**

Implementare una funzione **addVeicolo** che riceve come input una variabile di tipo **TcodaFIFO\* p** e variabile di tipo **Tveicolo v** ed inserisce il veicolo **v** nella lista (*put*). Nel caso di lista piena non eseguire alcuna operazione.

**[E - punti 2]**

Implementare una funzione **stampaVeicolo** che riceve in input una variabile di tipo **TcodaFIFO\* p** e stampa il contenuto della lista invocando il metodo **stampa**.

**[F - punti 6]**

Implementare una funzione **delAuto** che riceve in input una variabile di tipo **TcodaFIFO\* p** ed una cilindrata **c**: estrae (*get*) tutti gli elementi dalla coda e salva su file i soli veicoli di tipo AUTO che hanno il valore di **cil** < **c**.

Inserire i dati in un file denominato “**auto.txt**”, nella forma

**marca CILINDRATA=cil KW=kw**

È possibile creare strutture, funzioni, metodi aggiuntivi per realizzare gli esercizi proposti.

Se la funzione **non compila non verrà corretta!!**

**Il codice commentato NON verrà corretto!**